Analytical Modelling of Hot-Spot Traffic in Deterministically-Routed K-Ary N-Cubes

S. Loucif, M. Ould-Khaoua
Department of Computing Science
University of Glasgow, Glasgow, G12 8RZ, UK
{Mohamed/samia}@dcs.gla.ac.uk

G. Min
Department of Computing
University of Bradford, Bradford, BD7 1DP, UK
g.min@Bradford.ac.uk

Abstract

Many research studies have proposed analytical models to evaluate the performance of k-ary n-cubes with deterministic wormhole routing. Such models however have so far been confined to uniform traffic distributions. There has been hardly any model proposed that deal with non-uniform traffic distributions that could arise due to, for instance, the presence of hot-spots in the network. This paper proposes the first analytical model to predict message latency in k-ary n-cubes with deterministic routing in the presence of hot-spots. The validity of the model is demonstrated by comparing analytical results with those obtained through extensive simulation experiments.

1. Introduction

Wormhole routed *k*-ary *n*-cubes have been very popular interconnection networks for practical multicomputers [2, 8, 11, 16, 19] due to their desirable properties, such as ease of implementation, recursive structures, and ability to exploit communication locality to reduce message latency. Many routing algorithms have been suggested for wormhole routed *k*-ary *n*-cubes and can be widely classified as *deterministic* [5] or *adaptive* [7]. In deterministic routing messages always use the same path between a given pair of nodes, while in adaptive routing more flexibility is given to messages to choose their path in the network, avoiding congested regions and thereby reducing their latency. However, this flexibility is achieved at the expense of complex router hardware [1] in order to guarantee deadlock-freedom, due to the time to decide a route and the use of *virtual* channels; a virtual channel [3] has its own flit queue, but shares the bandwidth of the physical channel with other virtual channels in a time-multiplexed fashion. Moreover, recently, authors in [22] have shown that under realistic traffic patterns generated by typical parallel applications the performance advantages of deterministic routing can even approach those of adaptive routing without requiring complex routers.

Analytical models of both deterministic and adaptive routing in wormhole-routed networks, including *k*-ary *n*-cubes, have been widely reported in the literature [4, 6, 13,, 14, 18, 17]. However, these models have been based on the assumption that the traffic distribution across the network is uniform. The uniform traffic assumption is not always justifiable in practice as there are many parallel applications that exhibit non-uniform traffic patterns, which can produce, for example, hot-spots in the network [20]. Hot-spots arise when a number of nodes direct a fraction of their generated messages to a single destination node. There are several situations where this type of traffic distribution occurs. For instance, global synchronisation [23] where each node in the system sends a synchronisation message to a distinguished node is a typical situation that can produce hot-spots. Another example of hot-spot traffic can be found in the shared memory systems, where in some cache coherency protocols, to perform write-invalidation, a message is sent to all nodes having a dirty copy of the block. Those nodes, then, should send an acknowledgement back to the host node to maintain memory consistency correctly. So, if all nodes have a dirty copy of the block, this results in hot-spot traffic distribution.

Recently, analytical models of adaptive routing in hypercube [17] and torus [21] have been proposed, assuming non-uniform traffic distributions. Authors in [12] have proposed an analytical model of deterministic routing in hypercube with hot-spot traffic distribution. To the best of our knowledge, no study has been so far reported in the literature for modelling deterministic routing in high radix *k*-ary *n*-cubes in the presence of hot-spot traffic. Developing such a model would be very useful as most recent

Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)
1530-2075/05 $ 20.00 **IEEE**

practical machines have employed the 2D and 3D torus (instances of *k*-ary *n*-cubes with *n* =2 and 3). In an effort to fill this gap, the present paper suggests a new analytical model to predict message latency in deterministic-routed torus in the presence of hot-spot traffic.

The rest of the paper is organized as follows. Section 2 describes the *k*-ary *n*-cube. Section 3 outlines the analytical model. Section 4 validates the model through simulation experiments. Finally, section 5 concludes this paper.

2. The k-Ary n-Cube

The *k*-ary *n*-cube has $N = k^n$ nodes, arranged in *n* dimensions, with *k* nodes per dimension. Each node consists of a processing element (PE) and router, and is connected to its nearest neighbours in each dimension using bi-directional or uni-directional links. Our analysis considers only uni-directional case and can be easily extended to deal with bi-directional case. The router of a node is connected to its neighbouring nodes through *n* incoming channels, one for each dimension, and *n* outgoing channels. In addition, the router is connected to its local PE through injection and ejection channels, respectively; messages generated by the PE are injected into the network through the injection channel, and messages at the destination node are transferred to the PE through the ejection channel. The router contains flit buffers for each input virtual channel. The incoming and outgoing channels are connected by a crossbar switch, which can simultaneously connect multiple incoming to multiple outgoing channels given that there is no contention over the outgoing channels (see [17] for more description on the router structure in the *k*-ary *n*-cube).

3. Analysis

The present analysis considers only the 2-dimensional torus. Let us call these dimensions *x* and *y*, respectively. Furthermore, For the sake of clarity, let us consider the network as a set of *k* rings along each dimension, and let us call them *x*-rings and *y*-rings, respectively. Hot-spot messages that traverse *y* dimension use only one *y*-ring, which contains the hot-spot node. Let us refer to this ring as the "hot *y*-ring". Notice that regular messages, however, when crossing dimension *y*, can traverse channels of any *y*-ring, hot or non-hot *y*-ring, depending on source and destination location.

Let $v=(v_x, v_y)$ be the address of a given node in the network, other than the hot-spot node, and (v_{x_h}, v_{y_h}) be the address of the hot-spot node. A channel within the hot *y*-ring is said to be *j* $(1 \le j \le k)$ hops away from the hot-spot node when it is an outgoing channel from a node where $(v_{y_h} - v_y = j)$ if $v_{y_h} > v_y$, or $(k - v_y + v_{y_h} = j)$ otherwise. On the other hand, a channel within an *x*-ring is said to be *j* $(1 \le j \le k)$ hops away from the hot *y*-ring when it is an outgoing channel from a node where $(v_{x_h} - v_x = j)$ if $v_{x_h} > v_x$, or $(k - v_x + v_{x_h} = j)$ otherwise. A channel in the *y*-ring is *k* hops away from the hot-spot node when it is an outgoing channel from the hot-spot node itself, and a channel in the *x*-ring is *k* hops away from the hot *y*-ring when it is an outgoing channel from a node of the hot *y*-ring. Finally, an *x*-ring is said to be *j* $(1 \le j \le k)$ hops away from the hot-spot node when the nodes of that ring have $(v_{y_h} - v_y = j)$ if $v_{y_h} > v_y$, or $(k - v_y + v_{y_h} = j)$ otherwise; the *x*-ring that is *k* hops away from the hot-spot node is the one containing the hot-spot node itself.

The model is based on the following assumptions that have been widely used in previous network modeling studies [4, 6, 13, 17, 18, 21].

i) Nodes generate traffic independently of each other, and which follows a Poisson process with a mean rate λ messages/cycle.

ii) The traffic model proposed in [20] is used to generate hot spot traffic. In this model, each generated message has a finite probability h of being directed to the hot-spot node, and probability $(1-h)$ of being uniformly directed to the other network nodes.

iii) Message length is fixed and equal to L_m flits, each of which is transmitted through a physical channel in one cycle.

iv) The local queue at the injection channel in the source node has infinite capacity. Moreover, messages are transferred to the local PE as soon as they arrive at their destinations through an ejection channel.

v) Routing is deterministic where regular and hot-spot messages cross dimensions in a predefined order (without loss of generality, messages cross dimension *x* first then *y*).

vi) $V(=2)$ virtual channels are used per physical channel to avoid message deadlock in the torus due to the wrap-around channels [5].

The average number of channels that a regular message can cross within a dimension is given by

$$\bar{k} = \frac{\sum_{i=1}^{k-1} i}{k-1} = \frac{k}{2} \qquad (1)$$

The average number of channels crossed by regular messages within the network is

$$\bar{d} = n\bar{k} \qquad (2)$$

On the other hand, hot-spot messages can make, within a given dimension, from one to $(k-1)$ hops, and *i* $(1 \le i \le n(k-1))$ hops within the network.

The average regular traffic rate crossing channels of dimension x (dimension y respectively) is [21]

$$\lambda^r = \lambda(1-h)\bar{k} \quad (3)$$

Unlike regular traffic, hot-spot traffic is not uniformly distributed over channels. The calculation of hot-spot traffic rate on channels of dimension x (dimension y) requires the computation of the quantities: $P_{h_{x,j}}$ and $P_{h_{y,j}}$. $P_{h_{x,j}}$ represents the fraction of system nodes which generate hot-spot messages being routed on a channel along dimension x that is j hops away from the hot y-ring. Similarly, $P_{h_{y,j}}$ is the fraction of system nodes which generate hot-spot messages being routed on a channel along dimension y that is j hops away from the hot-spot node. $P_{h_{x,j}}$ and $P_{h_{y,j}}$ are found to be

$$P_{h_{x,j}} = (k-j)/N \quad (4)$$

$$P_{h_{y,j}} = k(k-j)/N \quad (5)$$

The fraction of the hot-spot traffic generated by system nodes and which crosses a channel of an x-ring, j hops away from the hot y-ring, respectively, a channel of the hot y-ring that is j hops away from the hot-spot node, $\lambda^h_{x,j}$ and $\lambda^h_{y,j}$, are given by

$$\lambda^h_{x,j} = N\lambda h P_{h_{x,j}} \quad (6)$$

$$\lambda^h_{y,j} = N\lambda h P_{h_{y,j}} \quad (7)$$

The total rate of regular and hot-spot traffic visiting a channel of an x-ring, respectively, a channel of the hot y-ring, $\lambda_{x,j}$ and $\lambda_{y,j}$, are expressed as follows

$$\lambda_{x,j} = \lambda^r + \lambda^h_{x,j} \quad (8)$$

$$\lambda_{y,j} = \lambda^r + \lambda^h_{y,j} \quad (9)$$

The mean message latency, *Latency*, is the sum of the mean waiting time of messages at the source and the time spent to cross the network (i.e., the mean network latency) scaled by the average degree of virtual channel multiplexing that takes place at a given physical channel. Taking $\overline{S^r}$ and $\overline{S^h}$ as regular and hot-spot message latencies, the mean message latency, *Latency*, is

$$Latency = (1-h)\overline{S^r} + h\overline{S^h} \quad (10)$$

Let us first focus on the calculation of the mean latency of regular messages. To determine $\overline{S^r}$, three cases are considered. The first case is when regular messages enter the network through dimension x, let $\overline{S^r_x}$ be the latency of this type of messages including the probability that these messages take that route. The second case is when regular messages enter the network through dimension y, skipping dimension x, and source and destination nodes belong to a non-hot y-ring. Let their mean message latency including the probability that they take that route be $\overline{S^r_{\bar{y}^h}}$. The last case is, also, when regular messages enter the network through dimension y but crossing the hot y-ring, let $\overline{S^r_{y^h}}$ be their message latency including the probability that messages take this path. So, $\overline{S^r}$ can be expressed as

$$\overline{S^r} = \overline{S^r_{y^h}} + \overline{S^r_{\bar{y}^h}} + \overline{S^r_x} \quad (11)$$

Regular messages crossing only the hot y-ring to reach their destination see at the entrance of the network a mean network latency $S^r_{y^h,\bar{k}}$, increased by their mean waiting time at the source Ws^r, both scaled by the average degree of virtual channels multiplexing $\overline{V}_{y^h}$. Thus $\overline{S^r_{y^h}}$ can expressed as

$$\overline{S^r_{y^h}} = \frac{1}{k(k+1)}\left(S^r_{y^h,\bar{k}} + Ws^r\right)\cdot\overline{V}_{y^h} \quad (12)$$

Similarly, the mean latency of regular messages crossing only a non hot y-ring, $\overline{S^r_{\bar{y}^h}}$, is given as

$$\overline{S^r_{\bar{y}^h}} = \frac{k-1}{k(k+1)}\left(S^r_{\bar{y}^h,\bar{k}} + Ws^r\right)\cdot\overline{V}_{\bar{y}^h} \quad (13)$$

Messages entering the network through dimension x see a mean network latency S^r_x and a mean waiting time at the source Ws^r, both increased by the multiplexing delay of virtual channels $\overline{V_x}$. $\overline{S^r_x}$ can be written as

$$\overline{S^r_x} = \left(S^r_x + Ws^r\right)\cdot\overline{V_x} \quad (14)$$

Furthermore, those messages can make their trip only in dimension x and see a mean network latency $S^r_{x,\bar{k}}$, or cross the hot y-ring once they exit dimension x and see a mean network latency $S^r_{x\to y^h,\bar{k}}$, or continue in a non-hot y-ring after crossing dimension x, in which case they see a mean network latency $S^r_{x\to\bar{y}^h,\bar{k}}$. Taking into account the three possible ways, S^r_x becomes

$$S^r_x = \frac{1}{k+1}\left(S^r_{x_t,\bar{k}} + \frac{k-1}{k}\left((k-1)S^r_{x\to\bar{y}^h,\bar{k}} + S^r_{x\to y^h,\bar{k}}\right)\right) \quad (15)$$

Let us now calculate the mean service times $S^r_{y^h,\bar{k}}$, $S^r_{\bar{y}^h,\bar{k}}$, $S^r_{x,\bar{k}}$, $S^r_{x\to y^h,\bar{k}}$, and $S^r_{x\to\bar{y}^h,\bar{k}}$. In general, the mean service time seen by a message, regular or hot-spot, at the entrance of a channel is the sum of three components, notably its transfer time through the physical channel (one cycle time for the header), the

mean blocking delay experienced at that channel, and the mean service time at the entrance of the next channel. A regular message is at the j^{th} channel $(1 \le j \le \bar{k})$ when j channels are left to visit in that dimension. A regular message traversing only dimension y, and crossing a non-hot y-ring, may experience, at each physical channel, a blocking delay, $B\left(\lambda^r, 0, S^r_{\bar{y}^h,\bar{k}}, 0\right)$, resulting in

$$S^r_{\bar{y}^h,j} = 1 + B\left(\lambda^r, 0, S^r_{\bar{y}^h,\bar{k}}, 0\right) + \begin{cases} L_m & j = 1 \\ S^r_{\bar{y}^h,j-1} & 1 < j \le \bar{k} \end{cases} \tag{16}$$

In a similar way, $S^r_{y^h,j}$ is found to be

$$S^r_{y^h,j} = 1 + (1/k)\sum_{l=1}^{k} B\left(\lambda^r, \lambda^h_{y,l}, S^r_{y^h,\bar{k}}, S^h_{y,l}\right) + \begin{cases} L_m & j = 1 \\ S^r_{y^h,j-1} & 1 < j \le \bar{k} \end{cases} \tag{17}$$

Note that in the hot y-ring, blocking at a given channel is due to the contention with both regular and hot-spot messages. A channel can be l $(1 \le l \le k)$ hops away from the hot-spot node with the probability $1/k$, in which case the hot-spot traffic rate crossing that channel is $\lambda^h_{y,l}$, and requires a mean service time $S^h_{y,l}$. So, the average blocking delay seen by a regular message is taken as the average of blocking delays over all k channels of the hot y-ring.

Regular messages visiting only dimension x see a mean service time at the entrance of the first channel, $S^r_{x,\bar{k}}$, and in general they see at the j^{th} $(1 \le j \le \bar{k})$ channel a mean service time $S^r_{x,j}$. Similarly, regular messages, which continue their trip in a non-hot y-ring, see, at the entrance of the first channel of dimension x, a mean service time $S^r_{x\to\bar{y}^h,\bar{k}}$. Finally, regular messages which continue their trip in the hot y-ring after crossing dimension x, see at the entrance of the first channel of dimension y a mean service time $S^r_{x\to y^h,\bar{k}}$.

When regular messages traverse a channel of dimension x, the latter can be within any x-ring, t hops away from the hot-spot node with the probability $1/k$. Within the ring itself, that channel can be l hops away from the hot y-ring with the probability $1/k$. Moreover, a regular message entering a channel of dimension x may compete to acquire that channel with other regular messages of rate λ^r, and hot-spot messages of rate $\lambda^h_{x,l}$. So, to find the blocking delay of regular messages at a given channel of dimension x, the latter is taken as the average of blocking delays over all channels of x-rings. The mean network latencies $S^r_{x,\bar{k}}$, $S^r_{x\to\bar{y}^h,\bar{k}}$, and $S^r_{x\to y^h,\bar{k}}$ can be written, then, as

$$S^r_{x,j} = 1 + (1/k^2)\sum_{t=1}^{k}\sum_{l=1}^{k} B\left(\lambda^r, \lambda^h_{x,l}, S^r_{x,\bar{k}}, S^h_{xl,t}\right) + \begin{cases} L_m & j = 1 \\ S^r_{x,j-1} & 1 < j \le \bar{k} \end{cases} \tag{18}$$

$$S^r_{x\to\bar{y}^h,j} = 1 + (1/k^2)\sum_{t=1}^{k}\sum_{l=1}^{k} B\left(\lambda^r, \lambda^h_{x,l}, S^r_{x,\bar{k}}, S^h_{xl,t}\right) + \begin{cases} S^r_{\bar{y}^h,\bar{k}} & j = 1 \\ S^r_{x\to\bar{y}^h,j-1} & 1 < j \le \bar{k} \end{cases} \tag{19}$$

$$S^r_{x\to y^h,j} = 1 + (1/k^2)\sum_{t=1}^{k}\sum_{l=1}^{k} B\left(\lambda^r, \lambda^h_{x,l}, S^r_{x,\bar{k}}, S^h_{xl,t}\right) + \begin{cases} S^r_{y^h,\bar{k}} & j = 1 \\ S^r_{x\to y^h,j-1} & 1 < j \le \bar{k} \end{cases} \tag{20}$$

Depending on the source and hot-spot node locations, hot-spot messages can take one of the two possible paths. They can make their trip only in the hot y-ring in which case they expect a mean latency including the probability that this case happens $\overline{S^h_y}$. They can visit an x-ring then the hot y-ring and in that case they expect a mean latency, given that they have taken that route, $\overline{S^h_x}$. The average latency over all possible paths taken by hot-spot messages, $\overline{S^h}$, is

$$\overline{S^h} = \overline{S^h_y} + \overline{S^h_x} \tag{21}$$

A hot-spot message generated by a node, j $(1 \le j < k)$ hops away from the hot-spot node, sees at its first channel a mean service time $S^h_{y,j}$, a mean waiting time at the source $Ws^h_{y,j}$, and an average multiplexing delay of virtual channels $V^h_{y,j}$. $\overline{S^h_y}$ is then found to be

$$\overline{S^h_y} = \frac{\sum_{j=1}^{k-1}\left(S^h_{y,j} + Ws^h_{y,j}\right)\cdot V^h_{y,j}}{N-1} \tag{22}$$

When the hot-spot message is at a channel j hops away from the hot-spot node, it competes with regular

traffic λ^r, which needs a service time $S^r_{y^h,\bar{k}}$, and hot-spot traffic $\lambda^h_{y,j}$, which requires a service time $S^h_{y,j}$. Once crossing that channel, the message expects a mean service time L_m in the case it reaches the hot-spot node, otherwise $S^h_{y,j-1}$. Thus, $S^h_{y,j}$ can be written as

$$S^h_{y,j} = 1 + B\left(\lambda^r, \lambda^h_{y,j}, S^r_{y^h,\bar{k}}, S^h_{y,j}\right) + \begin{cases} L_m & j = 1 \\ S^h_{y,j-1} & 1 < j < k \end{cases} \quad (23)$$

Hot-spot messages entering the network through dimension *x*, can be generated by any node *j* $(1 \le j \le k)$ hops away from the hot *y*-ring and within any *x*-ring *t* $(1 \le t \le k)$ hops away from the hot-spot node. Averaging the latency over all possible routes, $\overline{S^h_x}$ is given by

$$\overline{S^h_x} = \frac{\sum_{t=1}^{k}\sum_{j=1}^{k-1}\left(S^h_{x_{j,t}} + Ws^h_{x_{j,t}}\right)\cdot V_{x_{j,t}}}{N-1} \quad (24)$$

where $S^h_{x_{j,t}}$ is the mean network latency seen by a hot-spot message at the source, located within an *x*-ring *t* hops away from the hot-spot node and *j* hops away from the hot *y*-ring, $Ws^h_{x_{j,t}}$ is the mean waiting time at the source, and $V_{x_{j,t}}$ is the average multiplexing degree of virtual channels. To compute $S^h_{x_{j,t}}$, three cases are considered. The first case is when the hot-spot message is at the last channel within the *x*-ring, and the latter contains the hot-spot node. The second case is when the hot-spot message is at the last channel within an *x*-ring other than the one containing the hot-spot node. The last case is when the hot-spot message is crossing a channel that is not the last in the *x*-ring. So, $S^h_{x_{j,t}}$ is given by

$$S^h_{x_{j,t}} = 1 + B\left(\lambda^r, \lambda^h_{x,j}, S^r_{y^h,\bar{k}}, S^h_{x_{j,t}}\right) + \begin{cases} L_m & t = k, j = 1 \\ S^h_{y,t} & t \ne k, j = 1 \\ S^h_{x_{j-1,t}} & j \ne 1 \end{cases} \quad (25)$$

Having derived the equations of network and message latencies for both regular and hot-spot traffic. These are function of the mean blocking delays and are computed below. In general, a channel can be traversed by regular and hot-spot traffic rates α and β, respectively, and where the mean service time expected by each is S_α and S_β, respectively. If $P_b(\alpha, \beta, S_\alpha, S_\beta)$ is the probability that a message is blocked at that channel, and $w_c(\alpha, \beta, S_\alpha, S_\beta)$ is the mean waiting time to acquire that channel, the mean blocking delay can be written as [10]

$$B(\alpha, \beta, S_\alpha, S_\beta) = P_b(\alpha, \beta, S_\alpha, S_\beta) \times w_c(\alpha, \beta, S_\alpha, S_\beta) \quad (26)$$

The probability of blocking is the product of the total traffic rate crossing the channel and the mean service time is the weighted service time, taking into account the rate and the service time expected by each type of traffic. So, $P_b(\alpha, \beta, S_\alpha, S_\beta)$ is expressed as

$$P_b(\alpha, \beta, S_\alpha, S_\beta) = (\alpha + \beta)\left[\frac{\alpha}{\alpha+\beta} S_\alpha + \frac{\beta}{\alpha+\beta} S_\beta\right] \quad (27)$$

To determine the mean waiting time to acquire the channel, the latter is treated as an M/G/1 queue with a mean waiting time [6]

$$w_{\rho,S_v} = \frac{\rho S_v\left(1 + \frac{(S_v - L_m)^2}{{S_v}^2}\right)}{2(1 - \rho S_v)} \quad (28)$$

where ρ and S_v are the traffic rate and the mean service time at the channel, respectively. Using the above equation, the mean blocking delay at a channel, $w_c(\alpha, \beta, S_\alpha, S_\beta)$, becomes

$$w_c = \frac{(\alpha+\beta) S_{\alpha+\beta}\left(1 + \frac{(S_{\alpha+\beta} - L_m)^2}{S^2_{\alpha+\beta}}\right)}{2(1 - (\alpha+\beta) S_{\alpha+\beta})} \quad (29)$$

$$S_{\alpha+\beta} = \left(\frac{\alpha}{\alpha+\beta}\right) S_\alpha + \left(\frac{\beta}{\alpha+\beta}\right) S_\beta \quad (30)$$

Examining all above equations reveal that there are several interdependencies between the different variables of the model. Given that a closed-form solution to these interdependencies is very difficult to determine, the different variables of the model are computed using iterative techniques for solving equations [12, 17, 21].

With moderate to high traffic loads, messages are blocked at the source node before acquiring their fist network channel. The mean waiting time depends on the location of the source compared to the hot-spot node. The mean waiting time of regular messages at the source, Ws^r, is the average of waiting times at all system nodes. At any source, a regular message sees a mean network latency S_r

$$S_r = S^r_x + \frac{1}{k(k+1)}\left[S^r_{y^h,\bar{k}} + (k-1) S^r_{\bar{y}^h,\bar{k}}\right] \quad (31)$$

The mean network latency seen by a hot-spot message depends on how far that node is from the hot-spot node.

When the hot-spot message is generated by a source, in the hot *y*-ring and *j* hops away from the hot-spot node, sees a mean network latency $S^h_{y,j}$, and the overall network latency seen by regular and hot-spot messages at that node is $S_{y_j} = (1-h)S_r + hS^h_{y,j}$. However, when a hot-spot message is generated by a node that is not in the hot *y*-ring; say a node within an *x*-ring *t* hops away from the hot-spot node and *j* hops away from the hot *y*-ring, sees a mean network latency $S^h_{x_{j,t}}$. The overall network latency seen by regular and hot-spot messages at that node is $S_{x_{j,t}} = (1-h)S_r + hS^h_{x_{j,t}}$. When the source is the hot-spot node, only regular traffic is generated and the mean network latency seen by that traffic is S_r. Modelling the local queue at the source as an M/G/1 queue with a mean arrival rate λ/V (since the physical channel is split into V virtual channels), and using Equation (28), Ws^r is given as

$$Ws^r = w_{\lambda/V,S_r} + \frac{\sum_{j=1}^{k-1} w_{\lambda/V,S_{y,j}}}{k-1} + \frac{\sum_{t=1}^{k}\sum_{j=1}^{k-1} w_{\lambda/V,S_{x_{j,t}}}}{k(k-1)} \quad (32)$$

In a similar way, the mean waiting times of hot-spot messages at the source $Ws^h_{y,j}$ and $Ws^h_{x_{j,t}}$ are calculated by replacing ρ and S_v by their appropriate values in Equation (28), i.e., $w_{\lambda/V,S_{y,j}}$ and $w_{\lambda/V,S_{x_{j,t}}}$.

Finally, the latencies of the regular and hot-spot messages are affected by virtual channel multiplexing delay. The latter is function of the traffic rate crossing the physical channel and the mean service time at that channel. In general, given that the total traffic rate crossing a physical channel is λ_φ, with respective regular and hot-spot traffic rates λ^r_φ and λ^h_φ, and the mean service time $S_\varphi = \left(\lambda^r_\varphi/\lambda_\varphi\right)S^r_\varphi + \left(\lambda^h_\varphi/\lambda_\varphi\right)S^h_\varphi$, where S^r_φ and S^h_φ are the expected service time for regular and hot-spot traffic, respectively, the Markovian model in [3] yields the following probabilities

$$q_v = \begin{cases} 1 & v = 0 \\ q_{v-1}\lambda_\varphi S_\varphi & 0 < v < V \\ q_{v-1}(\lambda_\varphi / (1/S_\varphi - \lambda_\varphi)) & v = V \end{cases} \quad (33)$$

$$P_v = \begin{cases} 1 \Big/ \sum_{l=0}^{V} q_l & v = 0 \\ P_{v-1}\lambda_\varphi S_\varphi & 0 < v < V \\ P_{v-1}(\lambda_\varphi / (1/S_\varphi - \lambda_\varphi)) & v = V \end{cases} \quad (34)$$

The average degree of virtual channels multiplexing that takes place at a given physical channel is given by [3]

$$\overline{V}_\varphi = \frac{\sum_{v=1}^{V} v^2 P_v}{\sum_{v=1}^{V} v P_v} \quad (35)$$

Channels of non-hot *y*-rings are only crossed by regular traffic, i.e., $\lambda^r_\varphi = \lambda^r$, which require a mean service time $S_\varphi = S^r_\varphi = S^r_{\bar{y}^h,\bar{k}}$, and the average virtual channel multiplexing delay at those channels, $\overline{V}_{\bar{y}^h}$, is found using Equations (32-34). In the hot *y*-ring, however, the average multiplexing delay of virtual channels, $V_{y^h,j}$, at physical channel, *j* hops away from the hot-spot node, is computed taking λ^r and $\lambda^h_{y,j}$ as regular and hot-spot traffic rates, respectively, and each requires a mean service time $S^r_{y^h,\bar{k}}$ and $S^h_{y,j}$, respectively. The average multiplexing delay over all physical channels of the hot *y*-ring is, then, given as

$$\overline{V}_{y^h} = \frac{\sum_{j=1}^{k} V_{y^h,j}}{k} \quad (36)$$

Finally, the average degree of virtual channel multiplexing that takes place at a physical channel, within an *x*-ring *t* hops away from the hot-spot node and *j* hops away from the hot *y*-ring, $V_{x_{j,t}}$, is

$$\overline{V}_x = \frac{\sum_{t=1}^{k}\sum_{j=1}^{k} V_{x_{j,t}}}{k^2} \quad (37)$$

4. Model Validation

The proposed model has been validated through a discrete event simulator, operating at the flit level. Each simulation experiment was run until the network reached its steady state, that is, until a further increase in simulated network cycles does not change the collected statistics appreciably. The network cycle time in the simulator is defined as the transmission time of a single flit across a physical channel.

Extensive simulation experiments have been conducted to validate the model for different combinations of network sizes, message lengths, and hot-spot fraction *h*, and the general conclusions have been found to be consistent across all cases considered. For illustrative purposes, the model validation is shown for the following cases: network size $N = 256$ nodes; message lengths $L_m = 32$ and 100 flits; fraction of hot-spot traffic $h = 20\%$, 40% and 70%.

Figures 1 and 2 depict the mean message latency predicted by the model against simulation results. As can be seen, the figures reveal that the analytical model

predicts the mean message latency with a reasonable degree of accuracy when the network operates in the light and moderate load regions. However, there are small discrepancies in the results provided by the model and simulation when the network is under heavy traffic and approaches the saturation point. This is due to the approximations that have been made in the analysis to ease the model development (see Equation (28)). Nevertheless, we can conclude that the model produces latency results with a reasonable degree of accuracy in the steady state regions and it can be a practical evaluation tool for gaining insight into the performance behaviour of deterministic routing in *k*-ary *n*-cubes in the presence of hot-spot traffic.

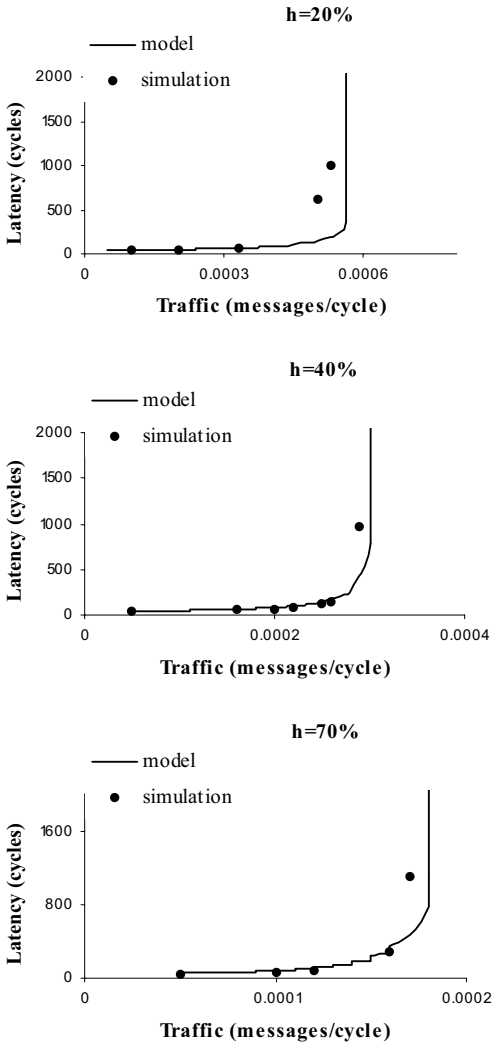

Figure 1. Latency predicted by the model against simulation results, L_m=32 flits.

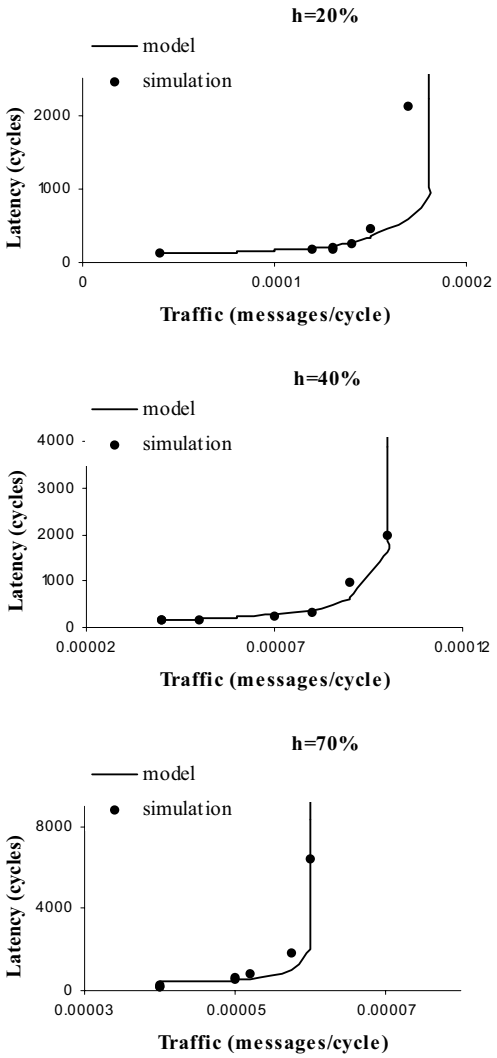

Figure 2. Latency predicted by the model against simulation results, L_m=100 flits.

5. Conclusions

This paper has proposed the first analytical model to predict the message latency of deterministic routing in the high radix *k*-ary *n*-cube in the presence of hot-spot traffic. The model has been validated through simulation experiments, and has been shown that it yields latency results which are in close agreement with those provided by simulations.

More recently, there have been some attempts to construct analytical models for interconnection networks operating under non-Poissonian traffic load, including bursty and self-similar traffic [16]. Our next objective is to extend the above modelling approach to deal with

such traffic patterns.

References

[1] A. A. Chien, "A cost and speed model for *k*-ary *n*-cube wormhole routers", *IEEE Trans. Parallel & Distributed Systems*, 9(2), 1998, pp. 150-162.

[2] Cray Research Inc., "The Cray T3E scalable parallel processing system", On Cray's Web http://www.cray.com/PUBLIC/productinfo/T3E/CRAY_T3E.html.

[3] W.J, Dally, "Virtual channel flow control", *IEEE Trans. Parallel and Distributed Systems*, 3(2), 1992, pp.194-205.

[4] W. Dally, "Performance analysis of k-ary n-cubes interconnection network", *IEEE Trans. Computers*, 39(6), 1990 pp. 775-785.

[5] W.J. Dally, C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks", *IEEE Trans. Computers,* 36(5), 1987, pp. 547-553.

[6] J.T. Draper, J. Ghosh, "A comprehensive analytical model for wormhole routing in multicomputer systems", *J. Parallel & Distributed Computing*, 32(2), 1994, pp. 202-214.

[7] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole-routed networks", *IEEE Trans. Parallel & Distributed Systems*, 4(12), 1993, pp. 1320-1331.

[8] M. Fillo, S.W. Keckler, W.J. Dally, N.P. Carter, A. Chang, Y. Gurevich, W.S. Lee, "The M-Machine multicomputer", *Int. Journal Parallel Programming*, 25(3), 1997, pp. 183-212.

[9] R.E. Kessler, J.L. Swarszmeier, "Cray T3D: A new dimension for Cray Research", *Proc. Compcon*, 1993, pp. 176-182.

[10] Kleinrock, L., Queueing systems, vol. 1, John Wiley, New York, 1975.

[11] J. Laudon, D. Lenoski, "The SGI Origin a ccNUMA highly scalable server", *Proc. ACM/IEEE 24th Int. Symp. Computer Architecture*, 1997, pp. 241-251.

[12] S. Loucif, M. Ould-Khaoua, "Modelling latency in deterministic wormhole-routed hypercubes under hot-spot traffic", *J. Supercomputing*, 27(3), 2004, pp. 265-278.

[13] S. Loucif, M. Ould-Khaoua, "Analysis of fully adaptive routing in wormhole-routed tori", *Parallel Computing*, 27(1), 1999, pp. 1477-1487.

[14] G. Min, M. Ould-Khaoua, "A new performance model for wormhole-switched *k*-ary *n*-cubes", *IEEE Trans. Computers,* 53(5), 2004, pp. 601-613.

[15] L.M. Ni , K. McKinley, "A Survey of wormhole routing techniques in direct networks", *IEEE Computers*, 26, 1993, pp. 62-76,.

[16] N Cube-2, N CUBE Company, N CUBE 6400 Processor Manual, 1990.

[17] M. Ould-Khaoua, H. Sarbazi-Azad, "An analytical model of adaptive wormhole routing in hypercubes in the presence of hot-spot traffic", *IEEE Trans. Parallel & Distributed Systems*, 12(3), 2001, pp. 283-288.

[18] M. Ould-Khaoua, "A performance model for Duato's adaptive routing algorithm in *k*-ary *n*-cubes", *IEEE Trans. Computers*, 48(12), 1999, pp. 1-8.

[19] C. J. Peterson, J. Sutton, P. Wiley, "iWARP: A 100-MPOS LIW microprocessor for multicomputers", *IEEE Micro*, 11(13), 1991, pp. 26-87.

[20] G.J. Pfister, V.A. Norton, "Hot-spot contention and combining in multistage interconnection Networks", *IEEE Trans. Computers*, 34(10), 1985, pp. 943-948.

[21] H. Sarbazi-Azad, L. Mackenzie, M. Ould-Khaoua, "Analytical modelling of wormhole-routed *k*-ary *n*-cubes in the presence of hot-spot traffic", *IEEE Trans. Computers*, 50(7), 2001, pp. 623-634.

[22] A.S. Vaidya *et al*, "Impact of virtual channels and adaptive routing on application performance", *IEEE Trans. Parallel & Distributed Systems*, 12(2), 2001, pp. 223-237.

[23] H. Xu *et al*," Efficient implementation of barrier synchronisation in wormhole-routed hypercube multicomputers", *J. Parallel & Distributed Computing*, 16, 1992, pp. 172-184.